**MICROPROCESSOR & EMBEDDED SYSTEMS (Project 1 & Part-2)**

**ALU, Register File & Multiplexors**

Today in the lab, we implemented an ALU, a register file, and a multiplexor using Verilog code with predetermined input and output, tested the module using a testbench, and verified the results using waveform output. as seen in the screenshots that are linked below.

In *image 1.1*

**OprdA** is Contains the data.

**OprdB** is the address that passes from the testbench.

**FS** is Various kind of Arithmetic Operation.

**SH** Represent the shift value or offset value.

**Fout** is output of arithmetic Operation between OprdA & OprdB.

**Z,C,V,N,D** is Flags which is set/clear on basic of Fout .

Graphical user interface

Description automatically generated

***Image 1.1***

In *image 1.2*

**AA,BA,DA** is the address that passes from the testbench.

**DataIn** is data that is successfully stored in Registers.

**addr & data** is a testbench variables given in testbench.

**WR** specifies the memory read or write operation.

In Our case RW = 1 for write and RW = 0 for Read.

**Reset** is 1 then Register file will be defaulted.

Graphical user interface

Description automatically generated with medium confidence

***Image 1.2***

In *image 1.3*

Is a Testing picture of 3:1 Mux.

Graphical user interface, application

Description automatically generated

***Image 1.3***

I would design and simulate a synthesizable Instruction Decoder and Constant Unit as part of my next week’s implementation strategy. There are numerous requirements and inputs listed that must be fed to them. Therefore, I am looking forward to working with Instruction Decoder and Constant Unit in the upcoming lab session on Thursday.